Ns-2 based Emulator for Multicast Services over WIMAX

Mayank Singh¹ Khan Fahad² Prof. Rajavikram Reddy. E³
¹,²,³Department of Electronics & Telecommunication Engineering
³M.H. Saboo Siddik College of Engineering Mumbai University, India

Abstract— Due to lack of availability of WiMax network, video transmission on WiMax can’t be achieved in real-time. In this paper we have presented a ns-2 based emulator to evaluate the quality of multicast video service transmission over WiMax (i.e. a wireless network). The result of different types of modulation schemes and the cyclic prefix, on the throughput and video quality were seen and the performance of the proposed emulator was enhanced comparatively. The results about the emulator's performance are of great help when real-time services are taken into account. The main reason of this paper is to lay guidelines required to develop an efficient real-time multicast streaming video testbed on wireless networks, it also helps us in exploring the potentialities of a tool such as ns-2

Key words: Ns-2, Wimax, Multicast Video Streaming Testbed

I. INTRODUCTION

There are various services such as Multicast and Broadcast Services (MBS) (which is also known as multimedia broadcast/multicast services, MBMS in 3GPP LTE) which require the support of an efficient wireless network which is no other than WiMAX network having an IEEE standard 802.16 [1]. MBS is a type of service where the transmission of packet data takes place from a single point source to multiple destination simultaneously. There are different algorithms through which we can get efficient emulated result, which will yield us the same result when employed in real-time. But the focus of any researcher should be mainly on resource management algorithm so that the spectral efficiency of MBS under a given condition i.e. quality of service (QoS). Since there is a lack of a well-operated WiMAX network, the output of the proposed algorithms can only be verified on the basis of computer simulations. The results obtained based on analysis or simulation are normally derived based on a single layer (e.g., physical or medium access control layer) point of view. Hence, the impact of the degraded QoS, which is normally represented in terms of packet loss rate in the analysis/simulation, the quality of the video which is received by client can’t be evaluated once it has reached it’s destination.

Many components may affect the performance of multimedia application. The components include users perception, multimedia applications (e.g. video codes), devices that operates the multimedia application, and network components that transport the multimedia content [2]. Mostly determination of QoS of any multimedia service can be done from networks or applications or users view. A certain level of quality is offered to multimedia application by using resource management, only if networking is the area to be considered. The quality level is normally determined in terms of network QoS parameters such as delay, packet loss rate, and/or throughput. However, the users are least concerned about how the management of resource is done by the network or what mechanisms are involved in improving the QoS. In reality, the users are concerned about the services that networks provide which has a direct impact on the perceived quality of the application.

There are many ways in which the user can find out the factors that are more responsible or highly influential for application’s QoS. Many requirements of the user can be fulfilled by using the application layer technique. This technique can adjust the frame size, frame rate, color depth, and the compression algorithm. One may also utilize various link/physical layer techniques such as packet scheduling, radio resource allocation, automatic retransmission request (ARQ), and forward error correction (FEC) to minimize the impact introduced by the wireless network.

Mostly, application-layer and link/physical-layer experts chose different metrics to indicate the quality of the multimedia services. For application-layer experts, a subjective mean opinion score (MOS) and a frame-by-frame peak signal to noise ratio (PSNR) are normally used as main performance metrics for sound/voice and video quality measurement, respectively. Basically, unreferenced or referenced methods are used to assess the quality of multimedia services [3]. Unreferenced methods evaluate the quality based on the degraded signal only. Subjective surveys are then required to map the degraded signal to MOS/PSNR. Referenced methods take into record, the distortion between original and degraded signals. The distortion values are then mapped to MOS/PSNR for the purpose of comparison. In any of the cases, the evaluation is done based on the real multimedia content and the quality is determined based on users feedback. For link/physical layer experts, QoS parameters of packet loss rate, delay, and delay jitter introduced by the network are chosen as performance metrics. Therefore, the evaluation is performed on individual multimedia packets and the quality is determined based on statistical results measured from simulation. Since we lack an integrated simulation/emulation platform, it is hard for the researchers to have a cross-layer optimization solution.

Several network architectures have been proposed for WiMAX emulator. Network being the concerned area some adopted the hardware-based approach and focused on measuring the physical layer signals. Other’s adopted a software-based approach but did not succeed in providing emulation results in real time. In most of the approaches, the emulation of single layer was presented. In [4], Chen proposed a general handover emulator for IEEE 802.16e system on top of NCTU_m network simulator. Four personal computers (PCs) are used to emulate two base stations (BSs), one mobile station (MSs), and one echo server. Only the ICMP echo packets and IEEE 802.16 MAC control frames are transmitted among these PCs. In [4], the author did not address issues of real-time processing and computational complexity since the size of the emulated packets is relatively small the traffic loading are insignificant. In [5,6], Mahrenholz and Ivanov investigated...
the real-time network emulation with ns-2. They tried to perform real-time emulation and interconnect ns-2 to a live network. In [4], a distributed architecture is adopted where each PC emulates as a single network node. In contrast, a centralized architecture is used in [5,6] where multiple network nodes are emulated by user-mode of Linux running on the same PC. The most useful benefit of well-distributed is that it can be easily scaled for various parameters. It is very hard to manage the PC’s which are used in setting up the architecture, since the transmission delay among the PC’s is very high. In contrast, the main benefit of the centralized architecture is easy for management. The scheduling accuracy and the computation power of the PC may affect the emulation results. Similar to [4,5,6], ICMP echo packets are exchanged among nodes to test the accuracy of real-time clock.

In this paper, an ns-2 based emulator is proposed to demonstrate the real performance of the multicast service. Similar to the approaches proposed by Mahrenholz and Ivanov [5,6], this paper used an ns-2 based WiMAX emulator to emulate different channel impairment experienced by each client in real-time. Different to their approaches, real streaming video packets are transmitted among nodes running on different PCs. Due to this, the video quality perceived by each client can be emulated in real-time. The performance of QoS mechanisms such as resource allocation and packet scheduling algorithms can then be evaluated from users point of view.

II. NETWORK ARCHITECTURE

The proposed network architecture is shown in Fig. 1. In this architecture, one PC is used as a media server and several PCs are used as video clients. One PC running ns-2 is used as a WiMAX network emulator. This depicts the impairment introduced in real-time by the wireless channel. The streaming video packets generated by the media server are fed in the WiMAX emulator and then multicast to clients. Each client can then display the received video stream independently.

The network simulator ns-2 is a discrete event simulator targeted at networking research. Ns-2 provides modules for simulation of TCP, routing, and multicast protocols over wired and wireless (local and satellite) networks. Ns-2 uses OTcl (Object oriented tool command language) for coding the program to run the simulation needed. This language is combination of two languages i.e. C++ & Tcl(Tool Command Language). It implements the main functions specified in IEEE 802.16-2004 [7] and IEEE 802.16e-2005 [8].

As shown in Fig. 1, each node in the WiMAX network is emulated by a ‘sink’ module plus two ‘wired node’ modules. It is because that the ‘wired node’ in ns-2 can only be configured as a Pcap/Berkeley Packet Filter (BPF) network object or a raw socket network object. The capturing and filtration of link-layer frames is done by Pcap/BPF. A raw socket network object is used to send network packets. For any node in WiMAX network, it can transmit and receive packets from the other nodes. Hence, two ‘wired node’ modules are needed to implement transmit/receive (TX/RX) functionality. A ‘sink’ module is used here to assign the TX/RX functionality to a BS or a MS.

III. REAL TIME VIDEO STREAMING

Finally, experiments are conducted to verify the effectiveness of the proposed WiMAX emulator. The network architecture used to conduct the experiments is shown in Fig. 2. Three PC were interconnected to a switch hub. The media server is equipped with a Pentium 4 CPU and Windows XP operation system. The WiMAX emulator is equipped with an AMD PhenomII X4 920 and RedHat Enterprise 5 operation system. The two clients are both equipped with an Intel Pentium M CPUs and Windows XP operation system. Since the RedHat software is not capable of to perform simulation on the windows platform, so the application changes the platform from windows to LINUX temporarily which acts as a virtual machine in the background.
IV. IMPLEMENTATION CHALLENGES

1) Implementation of video transmission from RTP media server to RTP media client.
2) Development of NS2 Emulator so that it can use external packets for simulation.
3) Implementation of WIMAX transmission in the NS-2 Emulator.

V. NEED FOR WIMAX

WiMAX is designed to replace the wired-broadband access networks while Wi-Fi was created for providing services into LAN networks.

At the physical layer, WiMAX channel sizes ranges from 1.75 MHz to 20 MHz while Wi-Fi based products require at least 20 MHz for each channel.

Wi-Fi uses the CSMA/CA (Carrier Sense Multiple Access with Collision Avoidance) which is not an efficient protocol. The MAC layer in WiMAX has been designed to scale from one to up 100s users within one RF channel.

In WiMAX, the base station assigns a QoS class to each connection. In 802.11, QoS was not considered in the early stage of its implementation.

WiMAX supports many transport technologies, such as ATM, IPv4, and IPv6 which are not supported by Wi-Fi.

WiMAX has the ability to support longer range transmission from 2 to 40 kilometers. While 802.11 was designed for low power consumption which limit the coverage to hundreds of meters.

Ns-2 is an event driven simulator. All of the events are stored in a queue and executed according to an ascending order of their trigger time. The task scheduler in Ns-2 always set Tt to be the trigger time of the first event until and unless the ongoing event is accomplished in the queue. In other words, the value of Tt is changed in a discrete manner. Normally, the difference between the trigger time of successive event is different from the time required to execute an event in CPU. Hence, there is a timing inconsistency between real and emulated network components. The timing inconsistency may lead to a serious problem when we interconnect the real and emulated network components. For example, let us assume that the media server transmits a streaming video packet every 1 second and the ns-2 takes 0.4 seconds to transmit and process a packet. In the media server, the first packet is generated at Tsys[1]. Ns2 receives the first packet immediately (i.e., Tsys[1]) and forwards the packet to client at Tsys[1.4]. i.e. a delay of 0.4 seconds is associated with it.

VI. EVALUATION

The following experiments have been conducted to verify the effectiveness of the proposed solutions. The accuracy of the real-time clock was first investigated. The performance of the floating-point operation and the fixed-point operation was then compared. The distortion of the fixed-point operation was also evaluated. Finally, the real-time streaming video was actually transmitted through the emulated network.

Fig. 4: Experimental Design

An experiment was designed to investigate the reduction of the computational complexity using the fixed-point operation. The round trip time required to transmit and receive an Internet control message protocol (ICMP) Echo packet was used as an index [9]. It is assumed that the round trip time comprises the transmission delay introduced by the network and the processing delay introduced by the CPU. The IP Table was set to ensure that the ICMP Echo packet will be processed by the WiMAX emulator and to prevent Linux kernel from responding ICMP Echo packets for the CPU.

VII. CONCLUSION

This paper has presented an ns-2-based IEEE 802.16 network emulator for supporting multicast streaming video services. The impact of the modulation schemes and the cyclic prefix, on the throughput and video quality has been discussed. Details of the implementation of the emulation in ns-2, including time synchronization, route coexistence problems and floating point accuracy have been addressed. Solutions have been proposed to create a link between video sender to a ns-2 emulator and transmit video over this emulated network to remote clients. The proposed emulator can emulate different channels and real propagation conditions. The video is displayed on each of the clients, and allows for evaluation of application-level video streaming metrics. This emulator can inject real traffic and can be used as a useful tool to test and develop new algorithms without using high-cost architectures.

The performance of the proposed emulator are evaluated when real-time services are taken into account. The main contribution in this paper is the general guidelines provided to develop an efficient real-time multicast video streaming testbed on wireless networks, exploiting the potentialities of ns-2 as a powerful tool.

REFERENCES


http://bmw.et.ntust.edu.tw/research/ns2emulation.asp